

CLUSTERING

EL PROCESO QUE SE CONOCE COMO CLUSTERING ES EL EJEMPLO MAS CLARO DE APRENDIZAJE NO SUPERVISADO, ES DECIR, QUE LOS DATOS NO DEBEN SER PREVIAMENTE ETIQUETADOS, NI SABER NADA DE ELLOS.

EL PROCESO CONSISTE EN, DADO UN SET DE DATOS SE QUIERE AGRUPAR LOS DATOS EN CLUSTERS DE FORMA TAL QUE TODOS LOS DATOS DENTRO DE UN MISMO CLUSTER SEAN SIMILARES ENTRE SI PERO DIFERENTES A LOS DE CUALQUIER OTRO CLUSTER.

CLUSTERING JERARQUICO

EL METODO DE CLUSTERING JERARQUICO GENERA UNA DESCOMPOSICIÓN JERARQUICA DEL SET DE DATOS CREANDO CLUSTERS DESDE $K=1$ HASTA M (SIENDO M LA CANTIDAD DE PUNTOS EN EL SET). UN PROBLEMA DEL CLUSTERING ES QUE NO ESCALA PARA GRANDES VOLUMENES DE DATOS.

EL ALGORITMO

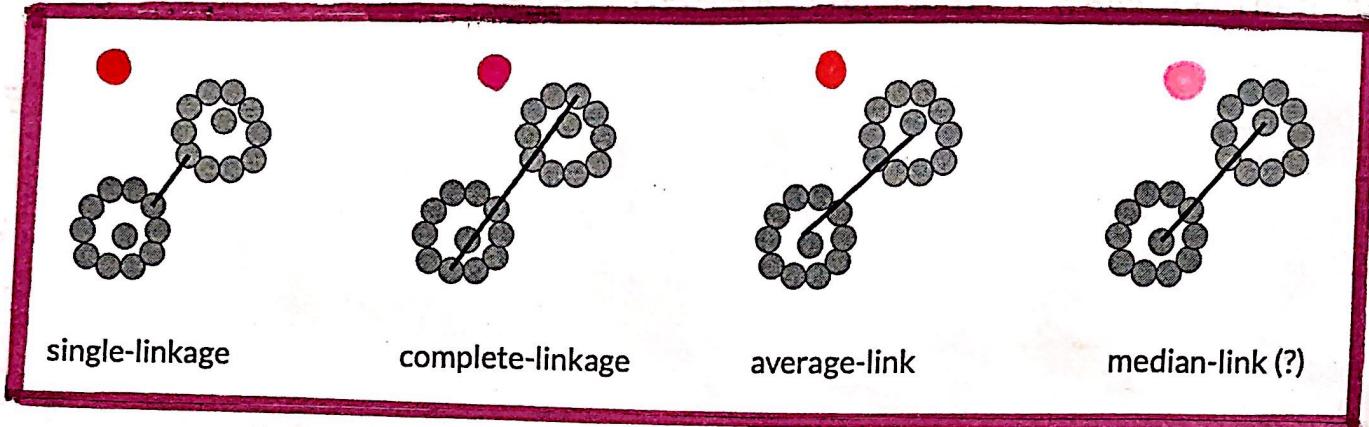
- COMENZAMOS SUPONIENDO QUE CADA PUNTO ES UN CLUSTER.
- 1 EN CONSECUENCIA EN ESTE PASO HAY M CLUSTERS.
- EN CADA PASO SE TOMAN LOS DOS CLUSTERS MAS CERCANOS.
- 2 ENTRE SI Y SE LOS UNE EN UN NUEVO CLUSTER.
- REPETIR EL PASO ANTERIOR HASTA QUE QUEDA UNICO CLUSTER.
- 3

QUE DISTANCIA UTILIZAS ENTRE DOS PUNTOS (\circ_1)?

ESTA PUEDE SER CUALQUIERA QUE SE AJUSTE A LA NATURALEZA DEL SET DE DATOS.

¿QUE DISTANCIA UTILIZAR ENTRE CLUSTERS (•z)?

AHORA NO ALCANZA CON COMPARAR PUNTO A PUNTO PORQUE UNO DE LOS CLUSTERS TIENE MAS DE UN PUNTO, ENTONCES VAMOS A MEDIR DISTANCIAS ENTRE GRUPOS:

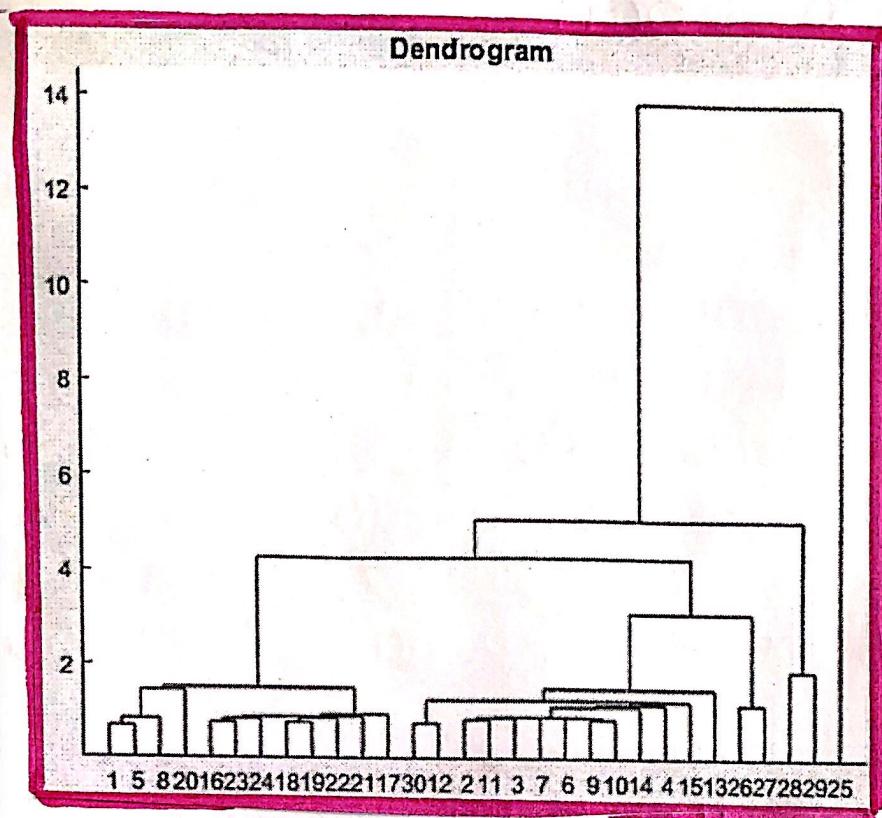


- LA DISTANCIA MINIMA ENTRE DOS PUNTOS DE CADA CLUSTER.
- LA DISTANCIA MAXIMA ENTRE DOS PUNTOS DE CADA CLUSTER.
- EL PROMEDIO DE LAS DISTANCIAS DE TODOS LOS PUNTOS DE UN CLUSTER CONTRA TODOS LOS PUNTOS DEL OTRO CLUSTER.
- LA DISTANCIA ENTRE LOS PROMEDIOS DE LOS PUNTOS DE CADA CLUSTER.

CANTIDAD DE CLUSTERS

LA CANTIDAD DE CLUSTERS A ELEGIR NO ES UNA TAREA TRIVIAL.

DENDROGRAMAS



A PARTIR DE LOS DATOS GENERADOS POR LA FUNCIÓN DE C.J., SE PUEDE GENERAR UN DIAGRAMA QUE NOS MUESTRE EN CADA PASO QUÉ CLUSTERS SE HAN UNIDO Y CON QUÉ DISTANCIA

- LA ALTURA DE LAS LÍNEAS REPRESENTA LA DISTANCIA A LA CUAL ESTABAN LOS PUNTOS.

EN ESTA FIGURA PODEMOS VER QUE $K=4$ ES MEJOR QUE $K=3$ (SIENDO K LA CANTIDAD DE CLUSTERS), YA QUE SE ESTABÍAN CORTANDO DISTANCIAS MÁS GRANDES. ESTO NOS DA UNA IDEA APROXIMADA DE CUÁLES PUEDEN SER LOS VALORES DE K MÁS ADECUADOS PARA NUESTRO SET.

PERFORMANCE

EN EL ALGORITMO ES NECESARIO EN CADA PASO ENCONTRAR LOS CLUSTERS MÁS CERCANOS, A DISTANCIA MÍNIMA **SINGLE-LINKAGE**. REALIZAR ESTO POR FUERZA BRUTA IMPLICA RECORRER LOS N CLUSTERS QUE HAY Y POR LO TANTO ESTO SERÍA $O(N^2)$.

ENTONCES LA SOLUCIÓN A ESTO ES UTILIZAR **LSH**. DE ESTA FORMA POR CADA CLUSTER SIMPLEMENTE SE BUSCA EL MÁS CERCANO ENTRE LA LISTA DE CANDIDATOS DEVUELTO POR LSH. ESTO PUEDE USARSE CALCULANDO LA DISTANCIA ENTRE CLUSTERS COMO LA DISTANCIA ENTRE CENTROÍDES. CADA VEZ QUE SE CREA UN NUEVO CLUSTER

SE PROMEDIAN SUS PUNTOS Y ESTE CENTROIDE SERÁ PUNTO QUE HASHEABA CON LSH. LUEGO POR CADA PUNTO SE CALCULA EL MAS CERCANO Y SE OBTIENE EL PAR DE DISTANCIA MINIMA. ESTO ES $O(N)$.

CLUSTERING K-MEANS

EL PROBLEMA DE CLUSTERING GENERAL RECIBE UNA SÉRIE DE M PUNTOS EN N DIMENSIONES. EL OBJETIVO ES ENCONTRAR COMO REPARTIR ESTOS M PUNTOS EN K CLUSTERS DE LA MEJOR FORMA POSIBLE. (K ES HIPER-PARAMETRO, SE SABE). LA 'MEJOR FORMA POSIBLE' ES AQUELLA QUÉ MINIMIZA LA DISTANCIA ENTRE CADA PUNTO Y EL CENTROIDE QUE SE LE HA ASIGNADO.

$$\min \sum_{i=1}^M \|x_i - c_i\|^2$$

BUSCAR LA POSICIÓN OPTIMA PARA $K=1$ ES SENCILLO, PERO LAMENTABLEMENTE A PARTIR DE $K=2$ EL PROBLEMA SE VUELVE COMPLEJO.

- ESTE PROBLEMA ES NP-HARD EN \mathbb{R}^2 , PARA CUALQUIER VALOR DE K .

AFORTUNADAMENTE EXISTEN HEURÍSTICAS MUY BUENAS QUE SON LAS CONOCIDAS COMO K-MEAN.

ALGORITMO DE LLOYD

ESTE ALGORITMO QUE ES EL CONOCIDO COMO K-MEANS, ES UNA FORMA DE APROXIMAR EL PROBLEMA DE CLUSTERING GENERAL.

ESTE ALGORITMO ES MUY SENCILLO YA QUE CONSTA DE UN ÚNICO CICLO Y DOS OPERACIONES.

EL ALGORITMO

- 1 INICIALIZO K CENTROIDES AL AZAR. EN ESTE CASO LOS CENTROIDES SON PUNTOS.
- 2 SE LE ASIGNA A CADA PUNTO EL CLUSTER CON CENTROIDE MAS CERCANO.
- 3 SE RECALCULAN LOS CENTROIDES COMO EL PROMEDIO DE LOS PUNTOS
- 4 SE REPITE EL PASO 2 Y 3 HASTA QUE LOS CENTROIDES CONVERGAN O SE LIMITE A UNA CIERTA CANTIDAD DE ITERACIONES

SE PUEDE GARANTIZAR QUE K-MEANS CONVERGE AL ANALIZAR LA FUNCION DE DISTORSION O COSTO

$$J(C, \text{IDX}) = \sum_{i=1}^M \|x_i - c_{\text{idx}(i)}\|^2$$

J ES MONOTONA Y DECRESCIENTE, LO CUAL NOS GARANTIZA SU CONVERGENCIA.

- LA CONCLUSION A ESTO ES QUE ESTE ALGORITMO ES MUY SENSIBLE A LA POSICION INICIAL DE LOS CENTROIDES. ESTO HACE QUE LA PERFORMANCE SEA MALA.

¿LA SOLUCIÓN? : K-MEANS ++

ESTA ES UNA VARIANTE DE K-MEANS EN DONDE LO UNICO QUE CAMBIA CON RESPECTO A LLOYD ES LA FORMA QUE SE INICIALIZAN LOS CENTROIDES. ESTA ASIGNA LOS CENTROIDES DE FORMA ESPACIADA.

EL ALGORITMO :

- 1 ELEGIR UN PUNTO AL AZAR COMO CENTROIDE
- 2 CALCULAR LA DISTANCIA DE CADA PUNTO CONTRA LOS CENTROIDES Y QUEDARSE CON LA MINIMA.
- 3 CALCULAR LA PROBABILIDAD DE CADA PUNTO, COMO LA DISTANCIA DIVIDIDA POR LA SUMA DE TODAS LAS DISTANCIAS DE LOS PUNTOS.

• ELEGIR UN NUEVO PUNTO AL AZAR, UTILIZANDO LA PROBABILIDAD CALCULADA

• REPETIR HASTA TENER k CENTROIDES.

→ EL PROBLEMA DE K-MEANS ++ ES QUE PARA DATOS REALMENTE MASIVOS NECESITA HACER k ITERACIONES SOBRE LOS DATOS PARA ELEGIR LOS CENTROIDES, ESTO ES MAS INEFICIENTE SI k ES UN NUMERO LARGO. PERO BUENO EN GENERAL ES UNA GRAN SOLUCIÓN Y ADEMÁS ACELEERA LA VELOCIDAD DE CONVERGENCIA.

¿DE DONDE SALE EL VALOR DE k ?

EN MUCHAS OCASIONES EL NÚMERO DE CLUSTERS (k) A GENERAR ES DESCONOCIDO. POR LO GENERAL SE USA GRID SEARCH PARA BUSCAR EL MEJOR VALOR.

ADEMÁS COMO METRICA PARA EVALUAR LA CALIDAD DE CADA VALOR DE k SE PUEDE USAR UN PROMEDIO DE NUESTRA FUNCION DE DISTORSIÓN PARA VARIAS ITERACIONES. HAY VARIAS METRICAS, HAY QUE OPTAR POR LA MAS FÁCIL. PARA PODER OBSERVAR EL k ÓPTIMO OBTENIDO CON LAS METRICAS, PODEMOS GRAFICAR Y NOS FIJAMOS DONDE SE GENERA EL 'CODO'.

DIAGRAMAS DE VORONOI

EL PROCESO DE CLUSTERING DE K-MEANS NOS GENERA ESTOS DIAGRAMAS. ESTE CONSISTE EN DELIMITAR LA FRONTERA HASTA LA CUAL LLEGA CADA CENTROIDE. ENTONCES ESTO NOS MUESTRA A QUÉ CLUSTER QUEDABA ASIGNADO UN PUNTO NUEVO SIN TENER QUE CORRER EL ALGORITMO. K-MEANS REALIZA UNA PARTICIÓN DE TODO EL ESPACIO N-DIMENSIONAL.

- CADA SECTOR DEL DIAGRAMA DEPENDE ÚNICAMENTE DEL CENTROIDE. Y EL TAMAÑO DE CADA ÁREA DEPENDE DE QUE TAN AISLADO ESTE EL CENTROIDE RESPECTO DEL DESTO. (+ AISLADO + GRANDE)

K-MEANS ONLINE

CUANDO HABLAMOS DE PROCESAR DATOS MASIVOS, EL PRINCIPAL PROBLEMA DE KM ES QUE DEQUIERE EN CADA ITERACIÓN CALCULAR EL CENTROIDE MAS CERCANO A CADA UNO DE LOS PUNTOS Y ESTO ES MUY COSTOSO. YA QUE HAY QUE VISITAR TODOS LOS PUNTOS DEL SET (α) POR CADA ITERACIÓN.

- LA VERSIÓN ONLINE PROCESA UN PUNTO POR VEZ DEL SET DE DATOS, NO NECESITA CONOCER TODOS LOS PUNTOS ANTES DE EMPEZAR Y DEQUIERE UN MINIMO USO DE MEMORIA.

EL ALGORITMO

- 1 INICIALIZA LOS K CENTROIDES DE ALGUNA FORMA EXTERNA AL ALGORITMO (AZAR).
- 2 SE PROCESA CADA PUNTO ASIGNANDOLE AL CENTROIDE MAS CERCANO.
- 3 SE MUEVE ESE CENTROIDE HACIA EL PUNTO PROCESADO DE FORMA PROPORCIONAL A LA CANTIDAD DE PUNTOS QUE TIENE EL CLUSTER.

HAY QUE DESTACAR QUE POR CADA PUNTO EL ALGORITMO COMPUTA LA DISTANCIA CONTRA CADA CENTROIDE PERO LA CANTIDAD DE CENTROIDES NUNCA ES MUY GRANDE. (EN CAMBIO KM COMPUTA EN CADA ITERACIÓN LA DISTANCIA DE TODOS LOS PUNTOS A TODOS LOS CENTROIDES).

→ CUANDO LOS PUNTOS PROVIENEN DE UN STREAM α ES NECESARIO Y SUFFICIENTE UNA UNICA ITERACIÓN.

A MEDIDA QUE SE PROCESAN MAS Y MAS PUNTOS, CADA PUNTO QUEDA ASOCIADO MEDIANTE SU CLUSTER A AQUELLOS QUE SON MAS CERCANOS - ESTO PERMITE REALIZAR CLUSTERING Y CLASIFICACIÓN ONLINE. SIN NINGUNA RESTRICCION DE TIEMPO NI MEMORIA.

LOS DIAGRAMAS VORONOI SUELEN SER MUY SIMILARES A LOS TRADICIONALES PERO ESTOS CONTIENEN ALGUNOS PUNTOS AISLADOS QUE QUEDAN DENTRO DE UN CLUSTER PERO PERTENECEN A OTRO. ESTO SUCEDE PORQUE ESEOS PUNTOS FUERON ASIGNADOS ORIGINALMENTE A UN CLUSTER CUYO CENTROIDE SE FUE MOVIENDO LUEGO. LA VERSIÓN ONLINE UNICAMENTE ASIGNA UNA VEZ POR PUNTO, Y ASÍ PERTENECE DENTRO DE DICHO CLUSTER.

ESTO PUEDE SOLUCIONARSE REALIZANDO MÁS DE UNA ITERACIÓN, ASÍ EN LA SEGUNDA ITERACIÓN LOS PUNTOS SE

REDUCCIÓN DE DIMENSIONES CON K-MEANS

SE PUEDE UTILIZAR K MEANS PARA REDUCIR LAS DIMENSIONES DE UN SET DE DATOS. UNA POSIBILIDAD ES REPRESENTAR A CADA PUNTO MEDIANTE SU CENTROIDE MÁS CERCANO. LOS SISTEMAS BASADOS EN COORDENADAS DE QUIEREN DE K-M PARA TENER SENTIDO. EL OBJETIVO ES ENCONTRAR UNA FORMA DE REPRESENTAR LOS DATOS EN K DIMENSIONES SIN QUE SEAN NECESARIOS LOS CENTROIDES.

• UNA FORMA SIMPLE DE HACER ESTO ES POR CADA PUNTO CALCULAR LA DISTANCIA A CADA UNO DE LOS K CENTROIDES Y USAR ESTAS K DISTANCIAS COMO K COORDENADAS POR CADA PUNTO. PARA CORREGIR PROBLEMAS DE ESCALA SE PUEDE USAR UNA FUNCIÓN QUE MAPEE LA DISTANCIA AL CENTROIDE X A UN NÚMERO ENTRE 0 Y 1. ($y = e^{-x^\gamma}$ SIENDO γ UN HIPO - PARÁMETRO)

→ ESTO NOS PERMITE EXTRAER K FEATURES A PARTIR DE UN SET DE DATOS EN CUALQUIER CANTIDAD DE DIMENSIONES LUEGO DE APLICAR K-MEANS CON K CENTROIDES. ES MUY COMÚN QUE UN ALGORITMO DE CLASIFICACIÓN FUNCIONE MEJOR CON ESTOS FEATURES QUE CON LOS ORIGINALES.

CLUSTERING ESPECTRAL

ESTE ALGORITMO TIENE LA CAPACIDAD DE DETECTAR FORMA DE CLUSTERS ARBITRARIAS, NO-LINEALES.

SEA $X \in \mathbb{R}^{m \times n}$ LA MATRIZ DE m DATOS EN n DIMENSIONES. COMO ESTE TRABAJA SOBRE COMPONENTES DE UN GRAFO VAMOS A UTILIZAR EL MISMO PROCESO QUE EN LAPLACIAN EIGENMAPS

• 1 CONSTRUCCIÓN DE LA MATRIZ DE AFINIDAD

$$W_{ij} = \exp - \frac{\|x_i - x_j\|^2}{2\sigma^2}$$

• 2 CONSTRUCCIÓN DE MATRIZ LAPLACIANA

$$L = D - W$$

LA MATRIZ LAPLACIANA DE UN GRAFO CONTIENE LA INFO SOBRE LA CONECTIVIDAD DE LOS \neq COMPONENTES.

→ SIENDO D LA MATRIZ DIAGONAL CON EL GRADO DE CADA VERTICE DEL GRAFO, DONDE POR GRADO SE ENTIENDE LA SUMA DE CADA COLUMNA w_{ii} .
 W ES LA MATRIZ DE ADYACENCIA DE UN GRAFO, EL GRADO ES LA CANTIDAD DE VERTICES CONECTADOS AL MISMO.

HAY VARIAS VARIANTES PARA CALCULAR EL LAPLACIANO.

• CIERTAS PROPIEDADES DE TEORIA DE GRAFOS DICEN QUE EL ESPECTRO DE LA MATRIZ LAPLACIANA SIRVE PARA ENCONTRAR CORTES MÍNIMOS EN UN GRAFO. UN CORTE MÍNIMO ES AQUEL QUE DIVIDE EL GRAFO EN DOS CONJUNTOS DE NODOS DE FORMA TAL QUE LAS AFINIDADES ENTRE NODOS DE LOS DOS CONJUNTOS TIENEN AFINIDAD MINIMA. ESTO ES EQUIVALENTE A REALIZAR CLUSTERING ENTRE PUNTOS QUE SE HAN REPRESENTADO COMO GRAFOS DEL NODO.

3 CALCULAR AUTOVALORES Y AUTOVECTORES

AHORA SE DESCOMPONE LA MATRIZ LAPLACIANA EN SUS AUTOVALORES Y AUTOVECTORES Y HAY QUE DETERMINAR LA CANTIDAD DE AUTOVECTORES A USAR. (ESTO SE PUEDE HACER ANALIZANDO EL PLOT DE AVAS). BUSCAMOS EL 'SALTO' EN EL PLOT. SI EL AUTOVALOR = 0 SU AVA LO IGNORAMOS. Y LA CANTIDAD DE AVAS QUE HAY ANTES DEL SALTO ES LA CANT CON LAS QUE REPRESENTO MIS DIMENSIONES.

→ FINALMENTE UNA VEZ QUE TENGO EL NUMERO DE AUTOVALORES A USAR, APLICO K-MEANS A LOS AUTOVEC Y ASIGNO LOS PUNTOS ORIGINALES DE ACUERDO A K-MEANS.

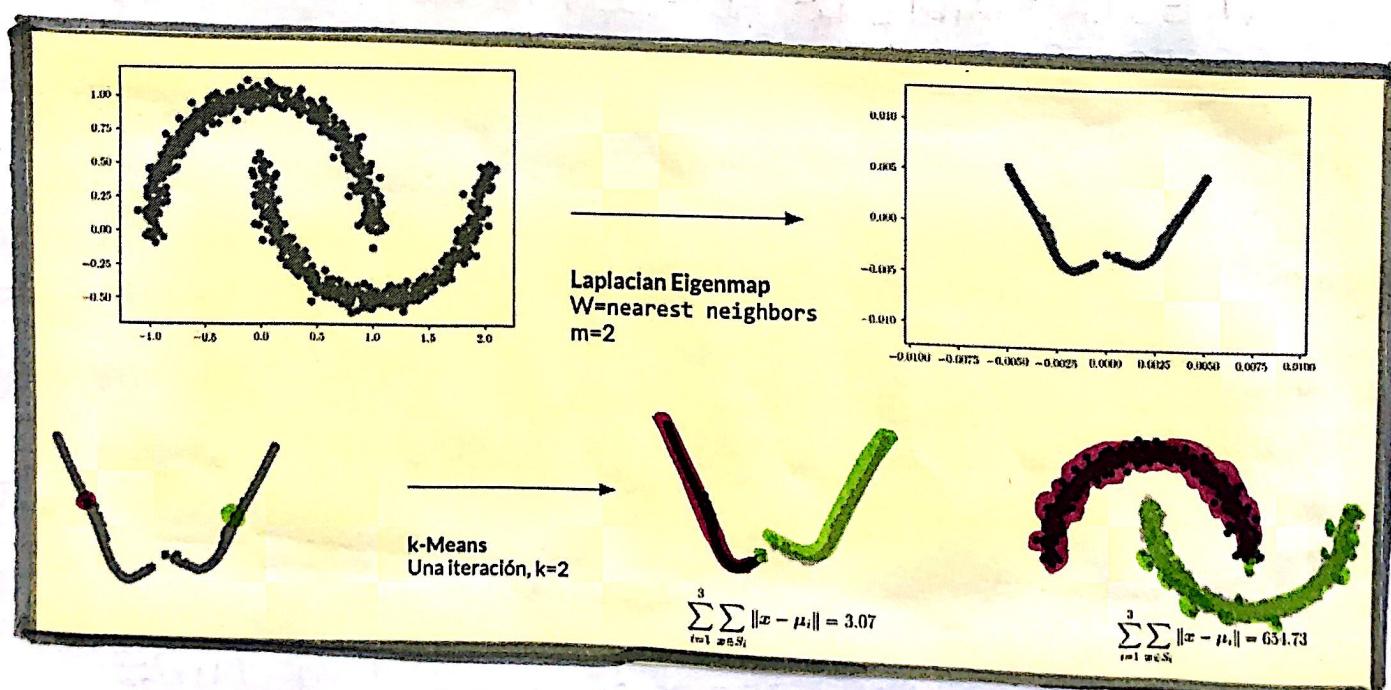
HIPER-PARAMETROS

- Γ EN LA MATRIZ AFINIDAD (W)

- LA FORMULA A UTILIZAR DE L .

- LA CANTIDAD DE AUTOVECTORES A USAR.

LA SELECCION DE ESTOS ES MUY CRITICA EN EL ALGORITMO.



DBSCAN

ESTE ALGORITMO ESTA BASADO EN EL CONCEPTO DE DENSIDAD. ENCUENTRA ZONAS EN LAS CUALES HAY UNA GRAN CANTIDAD (DENSIDAD) DE PUNTOS Y IDENTIFICA ESTAS ZONAS COMO CLUSTERS. DE ESTA MANERA DBSCAN ES CAPAZ DE DETERMINAR LA CANTIDAD DE CLUSTERS Y OUTLIERS (PUNTOS QUE NO PERTENECEN A NINGUN CLUSTER Y HACEN BUDIO).

HÍPER-PARÁMETROS

- ϵ : LA DISTANCIA MINIMA ENTRE DOS PUNTOS PARA ESTAR EN UN MISMO CLUSTER.
- k : LA CANTIDAD MINIMA DE PUNTOS VECINOS A LA DISTANCIA CORRECTA PARA PODER GENERAR NUEVOS CLUSTERS.

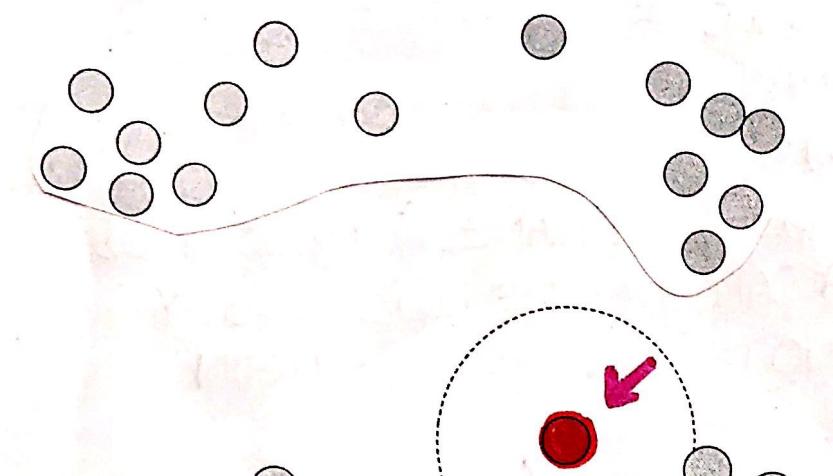
ENTONCES UN PUNTO ESTA EN UNA REGION DENSA SI Y SOLO SI HAY MAS DE k PUNTOS CON DISTANCIA MENOR A ϵ DEL PUNTO.

→ EL ALGORITMO COMIENZA CON UN PUNTO CUALQUIERA SI EL PUNTO NO PERTENECE A UN CLUSTER ENTONCES SE ANALIZA SI TIENE AL MENOS $k-1$ VECINOS A DIST. MENOR O IGUAL A ϵ EN CUYO CASO SE FORMA UN NUEVO CLUSTER CON k PUNTOS. SI EL PUNTO YA PERTENECE A UN CLUSTER ENTONCES SE BUSCAN LOS PUNTOS A DIST. ϵ O MENOR Y SE LOS AGREGA AL CLUSTER.

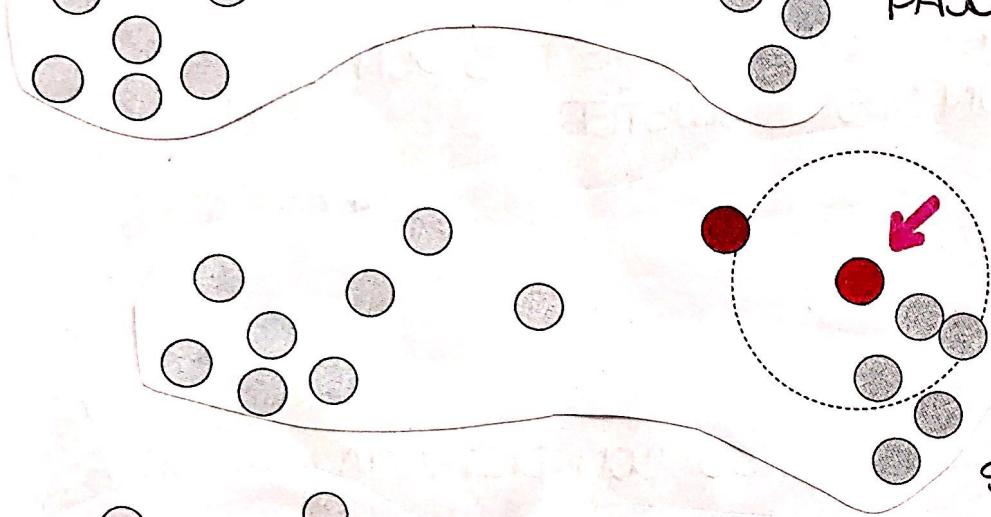
CUANDO LOS CLUSTERS TIENEN DISTINTAS DENSIDADES DBSCAN FALLA YA QUE NO ENCUENTRA UN ϵ QUE SIRVA. ENTONCES PUEDE ENCONTRAR GRUPOS DE CUALQUIER FORMA SIEMPRE Y CUANDO SE MANTenga LA DENSIDAD.

EJEMPLO

COMIENZO CON ESTA SERIE DE PUNTOS. DEFINIMOS $K=5$ Y E.

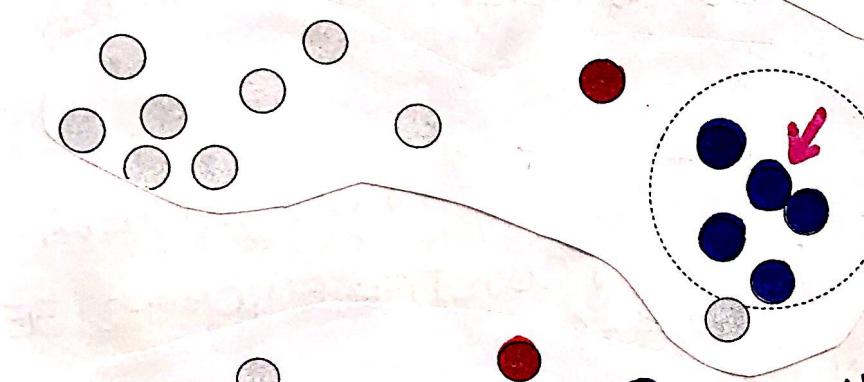


DE ESTOS PUNTOS AGREGO UNO AL AZAR Y TRASO EL 'RADIO' DE E PARA VER SI FORMA CLUSTER O NO.

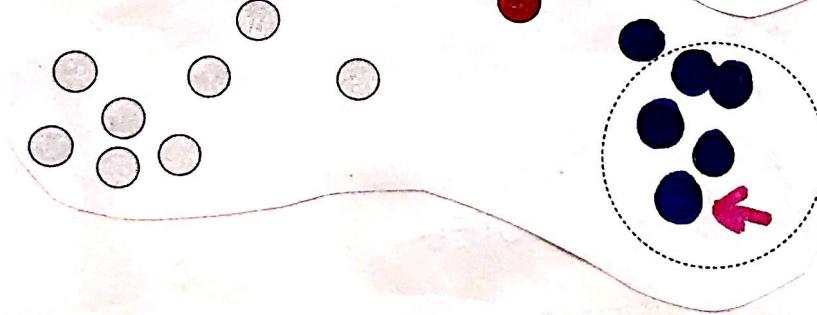


COMO EN ESTE PERIMETRO K ES MENOR A 5, MARCO EL PUNTO COMO UN OUTLIER Y PASO A OTRO.

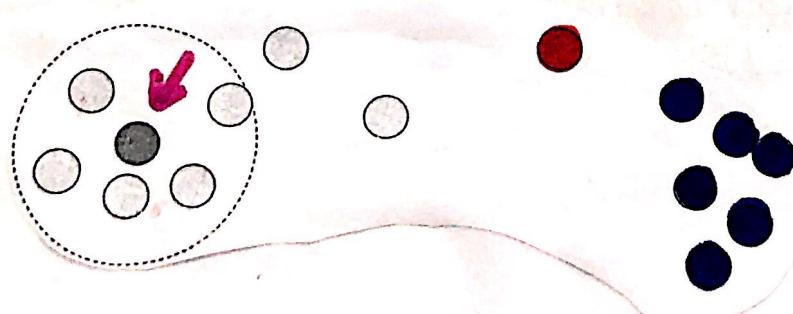
SUCede LO MISMO, ($K < 5$) ENTONCES LO MARCO COMO OUTLINED Y PASO A OTRO PUNTO.



CON ESTE PUNTO SUCede QUE $K=5$, ENTONCES FORMO MI PRIMER CLUSTER. (NOTAR QUE EL PREVIO MARCADO COMO OUTLINED AHORA ES DE CLUSTER)



UNA VEZ QUE FORMAMOS UN CLUSTER EXPLORAMOS A LO ANCHO.
(LOS PUNTOS QUE YA FUERON DEVISADOS EL METODO SE ACUERDA)



AGREGAMOS CADA PUNTO DE LA ZONA DENSA Y MIRAMOS LOS VECINOS. EN ESTE CASO ENCONTRAMOS UNO QUE PERTENECE AL AZUL

SEGUIMOS CON EL RESTO DE LOS PUNTOS.

AL AGREGAR ESTE PUNTO VEMOS QUE SE FORMO OTRO CLUSTER ASÍ QUE VOLVEMOS A EXPLODAR A LO ANCHO.

AL AGREGAR ESTE PUNTO VEMOS QUE SE ENCIERBAN MENOS DE 5 PUNTOS, ENTONCES EL VECINO QUEDA COMO OUTLI.

FINALMENTE EL ULTIMO PUNTO QUEDA COMO OUTLINED Y EL RESULTADO FINAL ES 2 CLUSTERS Y 3 OUTLINES.

DESVENTAJAS

- ES MUY SENSIBLE A LOS PARAMETROS, EN PARTICULAR
- NO MANEJA BIEN DENSIDADES DISTINTAS
- COMO ES 'PLANO' NO PUEDE ENCONTRAR CLUSTERS MÁS DENSOS DENTRO DE OTROS MÁS GRANDES.

→ ¿PORQUÉ DBSCAN SI YA TENGO CLUSTERING ESPEC.? DBSCAN ES MUCHO MÁS RÁPIDO !!

HDBSCAN

ESTE ALGORITMO ESTA BASADO EN DBSCAN, SOLUCIONA EL PROBLEMA DE LAS DENSIDADES DIFERENTES EN LOS CLUSTERS. EL PROBLEMA SUCEDE CUANDO SE USA UN ϵ MUY PEQUEÑA, LOS CLUSTERS CON DENSIDADES DE PUNTOS BAJAS QUEDAN SIN IDENTIFICAR (MUCHOS OUTLIERS). Y SI SUCEDE AL REVERZ LOS CLUSTERS MENOS DENSOS EMPIEZAN A MERGEAR ENTRE SÍ.

EL ALGORITMO

1) CONSTRUCCIÓN DE MATRIZ DE DISTANCIAS. ENTRE PUNTO UTILIZANDO LA DISTANCIA MBD.

MRD → SE ASOCIA A CADA PUNTO CON LA DISTANCIA DE SU VECINARIO, QUE ES LA DISTANCIA MAXIMA ENTRE SUS k VECINOS MAS CERCAOS. (k ES UN H-P)
ESTA DISTANCIA ES **CORE_K(x_i)**

$$MRD(A, B) = \max \{ CORE_K(A), CORE_K(B), D(A, B) \}$$

2) CONSTRUCCIÓN DE ÁRBOL MÍNIMO

USANDO LA MATRIZ DE DISTANCIAS MRD, EL OBJETIVO ES ENCONTRAR LAS 'ISLAS' DE PUNTOS QUE TIENEN ALTA DENSIDAD, EN DONDE LA DENSIDAD ES VARIABLE DENTRO DE CADA ISLA. CONSIDERAMOS A LOS DATOS COMO UN GRAFO EN DONDE EL PESO DE LAS ARISTAS ES LA DIST MBD ENTRE PUNTOS. CON ESTO CONSTRUIMOS UN ÁRBOL MÍNIMO PARA EL GRAFO. ESTO SE PUEDE HACER MEDIANTE K-MUSKAL.

SE EMPIEZA CON UN GRAFO VACÍO Y EN CADA PASO SE AGREGA LA ARISTA DE MENOR PESO. UNA VEZ QUE SE FORMA USAMOS CLUSTERING JEDABQUICO CON DIST MBD.

3) CLUSTERING JEDABQUICO

SI DECORDEMOS EL DENDROGRAMA DE ARRIBA HACIA ABAJO SE PARTE DE UN UNICO CLUSTER CON TODOS LOS PUNTOS QUE SE VAN SUBDIVIDIENDO EN OTROS CLUSTERS, AL IGUAL QUE EN DBSCAN SE QUIERE FIJAR UN TAMAÑO MÍNIMO PARA QUE UN CONJUNTO DE PUNTOS SE PUEDA CONSIDERAR CLUSTER.

ENTONCES PARA QUE UN SPLIT GENERE DOS NUEVOS CLUSTERS ESTOS TIENEN QUE TENER AL MENOS LA CANTIDAD MINIMA DE PUNTOS QUE SE ESTABLECIO, CASO CONTRARIO SE LO CONSIDERA COMO UN CLUSTER QUE REDDIO PUNTOS. ASI PODAMOS EL DENDROGRAMA.

1) EXTRAEER LOS CLUSTERS

INTUITIVAMENTE SE QUIEREN LOS CLUSTERS QUE PERDURAN A LO LARGO DEL TIEMPO (MIRANDO LOS DENDROGRAMAS PODADOS) LOS QUE DURAN POCO POSIBLEMENTE SON BUDIO.

QUEBREMOS, ENTONCES LOS DE MAYOR DURACION EN EL PLOT)

PARA MEDIR LA PERSISTENCIA DE LOS CLUSTERS NECESITAMOS:

- UNA MEDIDA DE DISTANCIA. PARA ELLA SE UTILIZADA $\lambda = 1 / \text{DISTANCIA}$. POR CADA CLUSTER SE TENDRAN DOS LAMBDA:
-
- ADEMÁS PARA CADA PUNTO p EN UN CLUSTER SE TIENE λ_p QUE ES EL VALOR DE λ EN EL MOMENTO EN EL CUAL EL PUNTO 'CAYO' FUERA DEL CLUSTER. ($\lambda_N > \lambda_p > \lambda_M$) ESTO NO QUIERE DECIR QUE p QUEDÓ FUERA DEL CLUSTER.

$$\text{ESTABILIDAD} = \sum_{p \in \text{CLUSTER}} (\lambda_p - \lambda_N)$$

ALGORITMO

EL ALGORITMO PARA EXTRAER LOS CLUSTERS COMIENZA CON CADA HOJA DEL DENDROGRAMA PODADO COMO UN CLUSTER SELECCIONADO. SE RECORRE EL ARBOL HACIA ARRIBA.

- SI LA SUMA DE LA ESTABILIDAD DE LOS DOS HIJOS ES MAYOR A LA ESTABILIDAD DEL PADRE ENTONCES AL PADRE SE LE ASIGNA LA ESTABILIDAD DE LOS HIJOS
- CASO CONTRARIO, SE DECLARA AL CLUSTER PADRE COMO UN CLUSTER SELECCIONADO Y SE DES-SELECCIONAN A LOS HIJOS

UNA VEZ QUE SE LLEGA A LA RAÍZ EL CONJUNTO DE CLUSTERS SELECCIONADOS CONSTITUYE EL GRUPO DE CLUSTERS.

CUALQUIER PUNTO QUE NO ESTE EN ALGUNO DE LOS CLUSTERS
ES RUIDO. RECORDAR QUE EL π_p DE CADA PUNTO EN UN
CLUSTER, LO PODEMOS USAR COMO LA PROBABILIDAD DE
QUE CADA PUNTO PERTENEZCA AL MISMO.

HDBSCAN TIENDE A DETECTAR MAS OUTLIERS DE LOS
NECESARIOS, DEBO ESTO PUEDE CORREGIRSE CONTROLANDO
EL δ Y VARIANDO LOS HIPER - PARAMETROS.